Brute-force, Dictionary Attacks and Mitigation

João Bettencourt

Fakulteta za elektrotehniko, računalništvo in informatiko, Univerza v Mariboru
Email: joao.bettencourt@um.si

The increasinng volume on digital systems in the modern society has increased the importance of security measures. One of the most persistent threats to digital security is the brute-force and dictionary attack, which exploits weak passwords to be harmful to systems. This paper explores these attack techniques, their mechanisms, and their impact on the modern society. By understanding the key points behind brute-force and dictionary attacks, we can identify vulnerabilities in password systems and propose effective mitigation strategies. This topic was chosen due to its significant relevance in today's digital age, where establishing security standards and practices is essential for protecting sensitive information. The study also emphasizes the need for stronger authentication methods and awareness to protect against evolving attack techniques. By combining theoretical analysis with practical demonstrations using tools such as Kali Linux, this work aims to highlight not only the risks posed by these attacks but also measures to improve digital security.

Keywords: Brute-Force Attacks, Dictionary Attacks, Password Security, Cybersecurity.

1. INTRODUCTION

Password-based authentication remains one of the most common methods for system security, based on their simplicity and effiency. However, it continues to show significant vulnerabilities, facing brute-force and dictionary attacks. Those methods exploit the inherent weaknesses of user-created passwords, which are often short, predictable and based on common patterns. In spite of decades of research and development of numerous countermeasures, the human factor remains the weakest link in the security trifecta.

By the new advancements in computational power, these vulnerabilities got aggravated, making attackers more effient cracking passwords. A task that used to require days now takes seconds or even less. At the same time, passwords cracking techniques have become progressively sophisticated, including vast dictionaries, permutations, and adaptive algorithms. These developments highlight the critical need for powerfull mitigation strategies that not only address technical vulnerabilities but also focus on improving user practices

This study investigates the mechanics of brute-force and dictionary attacks, demonstrates their effectiveness using practical experiments, and evaluates their strengths and weaknesses of most commonly used mitigation strategies. By analyzing passwords, by their entropy and characteristics, this work aims to emphasize the importance of adopting strategies to mitigate these attacks, by adopting stronger password policies and multi-factor authentication methods.

2. EXECUTION OF THE ATTACKS

Attacks...

2.1. Regular Expressions

Regular Expressions and What was used in this project

2.2. Brute-force Approach

How the Brute-force Approach happened

2.3. Dictionary attack Approach

How the Dictionary Approach happened

3. ANALYSIS

Analysis of what was done before

4. MITIGATION STRATEGIES

Short description of what is a mitigation strategie

4.1. Account Lockouts

Account lockout mechanisms are a security methods designed to restrain brute-force attacks by temporarily disabling accounts after a pre-agreed number of failed login attempts. This proceeds towards limit the number of guesses an attacker can make, by that means reducing the likelihood of unauthorized access. Nevertheless, while efficient mitigating brute-force attacks, they can still be exploited to launch denial-of-service (DOS) attacks, making legitimate useres unable to access their accounts. Research highlights that attackers can intentionally trigger account lockouts.[1] To make the security and usability in homeostasis, it is recommended to implement adaptive lockout policies that consider factors such as the user's typical behavior and the context of the login attempts. For instance, the Open Web Application Security Project (OWASP) suggests that while accounts are in lockout stage, they can prevent brute-force attacks, they should be configured to minimize the risk of being leveraged for DoS attacks [2] In conclusion, account lockout policies are a must have tool in defending against brute-force attacks but must be carefully configured to avoid unwanted consequences, such as DoS.

4.2. Captcha

CAPTCHAs, Completely Automated Public Turing tests to tell Computers and Humans Apart, are commonly used to distinguish human users from bots, thereby mitigating automatic attacks such as brute-force login attempts. By handing over challenges that are easy for humans but difficult for machines, CAPTCHAs can block bots from completing password guesses. However, CAPTCHAs are not absolute on the prevention of those attacks. Studies have shown that particular CAPTCHA systems, mainly those with low complexity or predictable patterns, can be vulnerable to automatic solving strategies. [3] As wel, the usability of CAPTCHAs is a critical consideration. The complexity of CAPTCHA can frustrate legitimate users, leading to a negative user experience, and can also make the system vulnerable. Therefore, while CAPTCHA can serve as a brake on brute-force attacks, balacing security needs with user accessibility. On top of that, CAPTCHAs should not be used as single and high-handed security measure, other methods should be used at the same time to provide a more robust security. [4]

4.3. Multi-factor Authentication

Multi-factor authentication most known as "MFA"enhances security by requiring useres to provide more than one form of verification, usually 2 to 3 ways of verification, before accessing a system. By combining something the user knows, something they have, or something they are, MFA adds layers of protection that forbid unauthorized access. Research indicates that implementing MFA can block over 99.9% of account compromising attacks, including brute-force and dictionary attempts.[5] Within the frame of brute-force attacks, where attackers widely attempt various password combinations to gain unauthorized acess, MFA serves as a tough brake. Even if an attacker is sucessfull guessing a password, the aditional verification factors required by MFA, prevent unauthorized entries. This multi-layered system ensures that the compromise of a single factor does not grant access, thus significantly enhancing security.[6] For that reason, adopting MFA is a crucial strategy to mitage risks associated with brute-force attacks, ensuring that only authorized userss can access sensitive systems and informations.

4.4. Password lenght

Password lenght is a critical factor to defend against brute-force attacks, where attackers can attempt multiple times all possible to estimate a password. The longer the passwords the longer it takes to be deciphered, increasing the number of potential combinations. Refering to a study analyzing real-world passwords found that shorter passwords are more receptive to brute-force attacks. with over 11% of them being cracked. [7] In addition , research indicates that a significant portion of compromised passwords are under 12 characters, suggesting that increasing password lenght can improve resistance to these attacks. [8] For that reason, enforcing longer password policies is an effective strategy to mitage the risk of digital break-ins.

4.5. SALT

Salting is a securtiy method that consists in adding a unique, random value as a šalt"to each user's password before hashing. This procedure guarantees that even identical passwords result in distinct hash outputs, mitigating efficiently the risk of precomputed attacks such as rainbow tables. By embracing salts, attackers are compelled to compute the has for each password attempt individually, notably increasing the computational effort required for a successful attack. Research indicates that salting, when hashing, empowers password security by preventing attackers from leveraging precomputed hash databases. [9] Besides, studies have exhibiting that salting, when hashing, provides strong defense against dictionary and brute-force attacks. The addition of salt to passwords increases the complexity of the hashed output, making it more challenging for attackers to crack passwords using pre assembled tables and systems. [10] For that reason, salting techniques are fundamental practices in enhancing password security and protecting against unauthorized access.

5. CONCLUSION

LITERATURE

[1] Account Lockouts: Characterizing and Preventing Account Denial-of-Service Attacks *Worcester Polytechnic Institute and Oak Ridge National Labo-*

ratory, https://web.cs.wpi.edu/~cshue/research/securecomm.19.lockouts.pdf.

[2] Blocking Brute Force Attacks *Esheridan*, https://owasp.org/www-community/controls/Blocking_Brute_Force_Attacks

[3] Practicality analysis of utilizing text-based CAPTCHA vs. graphic-based CAPTCHA authentication *Spring Nature Link*, https://pmc.ncbi.nlm.nih.gov/articles/PMC10152023/.

[4] CAPTCHA Types and Breaking Techniques: Design Issues, Challenges, and Future Research Directions *Cornell University*, https://arxiv.org/abs/2307.10239

[5] How effective is multifactor authentication at deterring cyberattacks? *Cornell University*, https://arxiv.org/abs/2305.00945.

[6] Multi-Factor Credential Hashing for Asymmetric Brute-Force Attack Resistance. *Cornell University*, https://arxiv.org/abs/2306.08169v1.

[7] Password Cracking with Brute Force Algorithm and Dictionary Attack Using Parallel Programming *MDPI*, https://www.mdpi.com/2076-3417/13/10/5979

[8] Do longer passwords protect you from compromise? *specops*, https://specopssoft.com/blog/longer-passwords-protect-compromise/

[9] Analysis of Password and Salt Combination Scheme To Improve Hash Algorithm Security *TheSai*, https://thesai.org/Downloads/Volume10No11/Paper_58-Analysis_of_Password_and_Salt_Combination_Scheme.pdf.

[10] Enhancing Salted Password Hashing Technique Using Swapping Elements in an Array Algorithm *ResearchGate*, https://www.researchgate.net/publication/352312164_Enhancing_Salted_Password_Hashing_Technique_Using_Swapping_Elements_in_an_Array_Algorithm.